UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
09/771,718	01/30/2001	Kenji Numata	045054/0135	9477

22428 7590 01/26/2005

FOLEY AND LARDNER
SUITE 500
3000 K STREET NW
WASHINGTON, DC 20007

EXAMINER
YIGDALL, MICHAEL J

ART UNIT	PAPER NUMBER
2122	

DATE MAILED: 01/26/2005

Please find below and/or attached an Office communication concerning this application or proceeding.

Office Action Summary	**Application No.** 09/771,718	**Applicant(s)** NUMATA, KENJI	
	Examiner Michael J. Yigdall	**Art Unit** 2122	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If the period for reply specified above is less than thirty (30) days, a reply within the statutory minimum of thirty (30) days will be considered timely.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on *01 September 2004*.

2a)☒ This action is **FINAL**. 2b)☐ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) *1-31* is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☐ Claim(s) _____ is/are allowed.

6)☒ Claim(s) *1-31* is/are rejected.

7)☐ Claim(s) _____ is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)☐ The specification is objected to by the Examiner.

10)☐ The drawing(s) filed on _____ is/are: a)☐ accepted or b)☐ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)☐ All b)☐ Some * c)☐ None of:

1.☐ Certified copies of the priority documents have been received.

2.☐ Certified copies of the priority documents have been received in Application No. _____.

3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) ☒ Notice of References Cited (PTO-892)
2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
3) ☐ Information Disclosure Statement(s) (PTO-1449 or PTO/SB/08)
Paper No(s)/Mail Date _____.
4) ☐ Interview Summary (PTO-413)
Paper No(s)/Mail Date. _____ .
5) ☐ Notice of Informal Patent Application (PTO-152)
6) ☐ Other: _____.

DETAILED ACTION

1. Applicant's response and amendment filed on September 1, 2004 has been fully considered. Claims 1-31 are pending.

Response to Arguments

2. Applicant's arguments have been fully considered but they are not persuasive.

3. Applicant contends that Yamauchi does not teach or suggest all of the features of the claimed state transition matrix (Applicant's remarks, page 38).

However, as Applicant acknowledges, Yamauchi discloses a state table and a transition table (see, for example, FIG. 3 and column 9, lines 26-39). Together, the state and transition tables form an operation specification table (see, for example, column 9, lines 26-28), which is analogous to a state transition table or a state transition matrix. Specifically, each item or "cell" of the table is defined by a state in which the system operates (such as the "start state" in FIG. 3) and an event corresponding to an input to the system (such as the "event" in FIG. 3). Each cell also includes information corresponding to a process to be executed by the system (such as the "action" in FIG. 3) and a transition destination state (such as the "end state" in FIG. 3).

4. Applicant contends that Yamauchi does not teach or suggest rewriting information for pseudo-generating an event stored in a memory section used in executing an event-generating routine into information corresponding to the event which is instructed to occur (Applicant's remarks, page 39).

However, Yamauchi discloses specifying a state transition model for a system, wherein the model includes states, events and transitions (see, for example, column 9, lines 11-25).

Yamauchi further discloses specifying, based on the model, a sequence of transitions that serves as a test sequence (see, for example, column 9, lines 48-56). The test sequence is stored in memory and includes information for generating events and performing the corresponding actions (see, for example, column 9, lines 57-64). Note that events generated while executing a test (see, for example, column 3, lines 62-67) are considered to be pseudo-generated events. Yamauchi further discloses outputting a test specification that is based, in part, on the test sequence (see, for example, FIG. 1). In other words, Yamauchi discloses rewriting the test sequence into a test specification, which is to say rewriting information for pseudo-generating events (such as the test sequence in FIG. 5) into information corresponding to the events (such as the test specification in FIG. 24).

5. Applicant contends that the claimed "script" features are not disclosed or suggested by Yamauchi (Applicant's remarks, pages 39-40).

However, Yamauchi discloses generating a test specification, as described above (see, for example, FIG. 24). The test specification is considered to be a script that describes the occurrence of events according to an input test sequence (see, for example, column 9, lines 48-64). Note that the tests account for the timing of each event (see, for example, column 4, lines 4-8). Yamauchi further discloses outputting the test specification or the script (see, for example, column 14, lines 14-25). Moreover, the states, events and transitions in the test sequence and in the test specification are output for display on a display section (see, for example, FIG. 4). The operation specification table, or the state transition matrix, as described above, is likewise output for display (see, for example, FIG. 2) according to position information (such as the "central coordinates" in FIG. 3).

6. Applicant contends that Yamauchi does not teach that the event is any one of a message-type, a flag-type, an interrupt-type, an in-mail type, and a function-call type, as recited in claims 12-14 (Applicant's remarks, page 40).

However, Yamauchi discloses that the events include internal events and external events (see, for example, column 3, lines 21-45). External events that cause state transitions in response to external operations (see, for example, column 10, lines 27-35) are considered to be interrupt-type events "for receiving an interrupt from an outside," as recited in the claims. Note that the claims recite that the event is merely "any one of" the above types, and therefore the claims do exclude the teachings of Yamauchi.

7. Applicant contends that the new limitations now recited in claims 9-11, as amended, are not disclosed or suggested by the combined teachings of Takuma and Yamauchi (Applicant's remarks, pages 40-41).

However, as presented below, Yamauchi discloses a main routine for executing main processes of the system and an event normal generating routine for, based on a signal supplied from the system due to operator selection performed on the system, detecting the operator selection and notifying said main routine of the operator selection. Yamauchi discloses these features in terms of an editor for detecting selections made by an operator and notifying the editor to update the display accordingly (see, for example, FIG. 2 and column 9, lines 11-25). The system inherently supplies signals based on the operator's interaction with the system, such as by moving the cursor.

Takuma in view of Yamauchi also discloses the limitation wherein said pseudo-generating routine is automatically generated in a programming language that is the same as or similar to a programming language of said main routine (see, for example, column 7, lines 37-50, which shows that the same VLIW processor executes both the instructions that compose the debug target program, i.e. the generated program, and the instructions that compose the main monitor program, i.e. the main routine, which is to say that the programming language is the same or is similar for both).

Claim Rejections - 35 USC § 102

8. The rejections of claims 1-31 under 35 U.S.C. 102(b) as being anticipated by Japanese Unexamined Patent Application Publication H06-175844 and by Republic of Korea Registered Utility Model Publication 0076805 are withdrawn in view of Applicant's remarks (pages 35-38) and the provided translations.

Claim Rejections - 35 USC § 103

9. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

10. Claims 1-31 are rejected under 35 U.S.C. 103(a) as being unpatentable over U.S. Pat. No. 6,141,791 to Takuma et al. (art of record, "Takuma") in view of U.S. Pat. No. 5,828,829 to Yamauchi et al. (art of record, "Yamauchi").

With respect to claim 1 (currently amended), Takuma discloses a program development apparatus used for developing a program to be installed in a system having at least a first central processing unit and an other component (see, for example, FIG. 2B, which shows a compiler and a debugger for program development on a system having a host processor and other components), said program development apparatus comprising:

(a) a program generating section for generating said program (see, for example, FIG. 11, which shows a compiler for generating the program).

Takuma does not expressly disclose an event pseudo-generating routine for pseudo-generating an event based on a state-transition matrix and event pseudo-generating information for pseudo-generating a same event as an event which normally occurs based on data or a signal transmitted from said other component to said first central processing unit in said system, wherein said state-transition matrix has a plurality of cells, each of said cells defined by: a) a state in which said system to be a subject of a program development is enabled to operate in, and b) an event which corresponds to an input to said system, and further wherein information corresponding to a process to be executed by said system and a transition state destination to be transited to when a corresponding event occurs during a corresponding state are stored for each said cell.

However, Yamauchi discloses the features above in a software development system used for testing programs (see, for example, the abstract). Yamauchi discloses causing or generating events based on state transitions (see, for example, column 4, lines 44-48, and column 3, lines 21-45, which shows that events are internal or external signals sent from a component to the processor). Note that events generated while executing a test (see, for example, column 3, lines

62-67) are considered to be pseudo-generated events. Yamauchi further discloses a state table and a transition table (see, for example, FIG. 3 and column 9, lines 26-39). Together, the state and transition tables form an operation specification table (see, for example, column 9, lines 26-28), which is analogous to a state transition table or a state transition matrix. Specifically, each item or "cell" of the table is defined by a state in which the system operates (such as the "start state" in FIG. 3) and an event corresponding to an input to the system (such as the "event" in FIG. 3). Each cell also includes information corresponding to a process to be executed by the system (such as the "action" in FIG. 3) and a transition destination state (such as the "end state" in FIG. 3).

It would have been obvious to one of ordinary skill in the art at the time the invention was made to incorporate the features taught by Yamauchi into the system of Takuma, so as to enable the testing of programs based on a plurality of events and state transitions (see, for example, Yamauchi, column 3, lines 62-67). The modification would have been obvious because one of ordinary skill in the art would have been motivated to facilitate debugging in the Takuma system (see, for example, Takuma, column 3, lines 33-36).

Takuma in view of Yamauchi also discloses:

(b) a second central processing unit having a same function as said first central processing unit and for executing emulation of said program and said event pseudo-generating routine (see, for example, FIG. 2B and column 8, lines 5-12, which shows an in-circuit emulator having a processor for performing the same functions as a first processor).

Although Takuma discloses emulating the program (see, for example, column 13, lines 54-63), Takuma does not expressly disclose:

(c) an analysis section for starting said emulation of said program from a state input as an initial state and for referring to said pseudo-generating information and rewriting information for pseudo-generating said event stored in a memory section used in executing said event pseudo-generating routine into information corresponding to said event which is instructed to occur.

However, Yamauchi further discloses specifying a state transition model for a system, wherein the model includes states, events and transitions (see, for example, column 9, lines 11-25). Yamauchi further discloses specifying, based on the model, a sequence of transitions that serves as a test sequence (see, for example, column 9, lines 48-56). The test sequence is stored in memory and includes information for generating events and performing the corresponding actions (see, for example, column 9, lines 57-64). Note that events generated while executing a test (see, for example, column 3, lines 62-67) are considered to be pseudo-generated events. Yamauchi further discloses outputting a test specification that is based, in part, on the test sequence (see, for example, FIG. 1). In other words, Yamauchi discloses rewriting the test sequence into a test specification, which is to say rewriting information for pseudo-generating events (such as the test sequence in FIG. 5) into information corresponding to the events (such as the test specification in FIG. 24).

With respect to claim 2 (currently amended), Takuma discloses a program development apparatus used for developing a program to be installed in a system having at least a first central processing unit and an other component (see, for example, FIG. 2B, which shows a compiler and a debugger for program development on a system having a host processor and other components).

Takuma does not expressly disclose:

(a) a state-transition matrix memory section for storing a state-transition matrix, wherein said state-transition matrix has a plurality of cells, each of said cells defined by: a) a state in which said system to be a subject of a program development is enabled to operate in, and b) an event which corresponds to an input to said system and further wherein information corresponding to a process to be executed by said system and a transition state destination to be transited to when a corresponding event occurs during a corresponding state are stored for each said cell;

(b) an event pseudo-generating editor for generating event pseudo-generating information for pseudo-generating a same event as an event which normally occurs based on data or a signal transmitted from said other component to a first central processing unit in said system.

However, Yamauchi discloses the features of (a) above in terms of a state table and a transition table (see, for example, FIG. 3 and column 9, lines 26-39). Together, the state and transition tables form an operation specification table (see, for example, column 9, lines 26-28), which is analogous to a state transition table or a state transition matrix. Specifically, each item or "cell" of the table is defined by a state in which the system operates (such as the "start state" in FIG. 3) and an event corresponding to an input to the system (such as the "event" in FIG. 3). Each cell also includes information corresponding to a process to be executed by the system (such as the "action" in FIG. 3) and a transition destination state (such as the "end state" in FIG. 3).

Yamauchi further discloses the features of (b) above in terms of an editor for defining a state transition model comprising events and actions (see, for example, FIG. 2 and column 9, lines 11-25). Note that events generated while executing a test (see, for example, column 3, lines 62-67) are considered to be pseudo-generated events.

It would have been obvious to one of ordinary skill in the art at the time the invention was made to incorporate the features taught by Yamauchi into the system of Takuma, so as to enable the testing of programs based on a plurality of events and state transitions (see, for example, Yamauchi, column 3, lines 62-67). The modification would have been obvious because one of ordinary skill in the art would have been motivated to facilitate debugging in the Takuma system (see, for example, Takuma, column 3, lines 33-36).

Takuma in view of Yamauchi also discloses:

(c) a program generating section for generating said program (see, for example, FIG. 11, which shows a compiler for generating the program).

Takuma does not expressly disclose an event pseudo-generating routine for pseudo-generating said event.

However, Yamauchi further discloses the features above in terms of generating events based on state transitions (see, for example, column 4, lines 44-48). Note that events generated while executing a test (see, for example, column 3, lines 62-67) are considered to be pseudo-generated events.

Takuma in view of Yamauchi also discloses:

(d) a second central processing unit for having a same function as said first central processing unit and for executing emulation of said program and said event pseudo-generating routine (see, for example, FIG. 2B and column 8, lines 5-12, which shows an in-circuit emulator having a processor for performing the same functions as a first processor).

Takuma does not expressly disclose:

(e) an input section for detecting which display position of each event or each state is indicated among a plurality of events and a plurality of states forming said state-transition matrix displayed on a display section and for outputting position information of said display position;

(f) an analysis section for converting said position information into an event code or a state code corresponding to said position so as to set a state corresponding to said state code as an initial state for starting emulation of said program and for referring to said pseudo-generating information so as to rewrite information stored in a memory section used in executing said pseudo-generating routine, said information for pseudo-generating an event into information corresponding to said event code.

However, Yamauchi further discloses the features of (e) above in terms of an input interface for defining and displaying a sequence of state transitions and events (see, for example, FIGS. 4, 6 and 8) and a state table or matrix having coordinates for a display position (see, for example, column 9, lines 26-30).

Yamauchi further discloses the features of (f) above in terms of converting a state transition model describing events and actions into a data structure, stored in memory, having event and state identification codes (see, for example, FIG. 3 and column 9, lines 11-39). Yamauchi further discloses specifying, based on the model, a sequence of transitions that serves as a test sequence (see, for example, column 9, lines 48-56), and outputting a test specification that is based, in part, on the test sequence (see, for example, FIG. 1). In other words, Yamauchi discloses rewriting the test sequence into a test specification, which is to say rewriting information for pseudo-generating events (such as the test sequence in FIG. 5) into information corresponding to the events (such as the test specification in FIG. 24).

With respect to claim 3 (currently amended), Takuma discloses a program development apparatus used for developing a program to be installed in a system having at least a first central processing unit and an other component (see, for example, FIG. 2B, which shows a compiler and a debugger for program development on a system having a host processor and other components).

Takuma does not expressly disclose:

(a) a state-transition matrix memory section for storing a state-transition matrix, wherein said state-transition matrix has a plurality of cells, each of said cells defined by: a) a state in which said system to be a subject of a program development is enabled to operate in, and b) an event which corresponds to an input to said system and further wherein information corresponding to a process to be executed by said system and a transition state destination to be transited to when a corresponding event occurs during a corresponding state are stored for each said cell.

(b) an event pseudo-generating editor for generating event pseudo-generating information for pseudo-generating a same event as an event which normally occurs based on data or a signal transmitted from said other component to a first central processing unit in said system.

However, Yamauchi discloses the features of (a) above in terms of a state table and a transition table (see, for example, FIG. 3 and column 9, lines 26-39). Together, the state and transition tables form an operation specification table (see, for example, column 9, lines 26-28), which is analogous to a state transition table or a state transition matrix. Specifically, each item or "cell" of the table is defined by a state in which the system operates (such as the "start state" in FIG. 3) and an event corresponding to an input to the system (such as the "event" in FIG. 3). Each cell also includes information corresponding to a process to be executed by the system (such as the "action" in FIG. 3) and a transition destination state (such as the "end state" in FIG. 3).

Yamauchi further discloses the features of (b) above in terms of an editor for defining a state transition model comprising events and actions (see, for example, FIG. 2 and column 9, lines 11-25). Note that events generated while executing a test (see, for example, column 3, lines 62-67) are considered to be pseudo-generated events.

It would have been obvious to one of ordinary skill in the art at the time the invention was made to incorporate the features taught by Yamauchi into the system of Takuma, so as to enable the testing of programs based on a plurality of events and state transitions (see, for example, Yamauchi, column 3, lines 62-67). The modification would have been obvious because one of ordinary skill in the art would have been motivated to facilitate debugging in the Takuma system (see, for example, Takuma, column 3, lines 33-36).

Takuma in view of Yamauchi also discloses:

(c) a program generating section for generating said program (see, for example, FIG. 11, which shows a compiler for generating the program).

Takuma does not expressly disclose an event pseudo-generating routine for pseudo-generating said event.

However, Yamauchi further discloses the features above in terms of generating events based on state transitions (see, for example, column 4, lines 44-48). Note that events generated while executing a test (see, for example, column 3, lines 62-67) are considered to be pseudo-generated events.

Takuma in view of Yamauchi also discloses:

(d) a second central processing unit for having a same function as said first central processing unit and for executing emulation of said program and said event pseudo-generating routine (see, for example, FIG. 2B and column 8, lines 5-12, which shows an in-circuit emulator having a processor for performing the same functions as a first processor).

Takuma does not expressly disclose:

(e) an input section for detecting which display position of each event or each state is indicated among a plurality of events and a plurality of states forming said state-transition matrix displayed on a display section so as to output position information of said display position and for generating an input event log including an order of instructed events and an instruction timing of each event.

(f) a script generating section for generating a script file in which an occurrence timing of each event and a timing at which an element in said system operates in accordance with a specification are descried based on said input event log.

(g) a script analysis section for sequentially outputting position information of each event described in said script file and of a corresponding display area in said state-transition matrix displayed on said display section in order and at an occurrence timing described in said script file.

(h) an analysis section for converting said position information into an event code or a state code corresponding to said position so as to set a state corresponding to said state code as an initial state for starting emulation of said program and for referring to said pseudo-generating information so as to rewrite information memorized in a memory section used in executing said pseudo-generating routine, said information for pseudo-generating an event into information corresponding to said event code.

However, Yamauchi further discloses the features of (e) above in terms of an input interface for defining and displaying a sequence of state transitions and events (see, for example, FIGS. 4, 6 and 8; note that FIG. 8 shows an event log output to a display; also see, for example, column 4, lines 4-8, which shows that the timing of the events is considered as well) and a state table or matrix having coordinates for a display position (see, for example, column 9, lines 26-30).

Yamauchi further discloses the features of (f) and (g) above in terms of generating a test specification or script (see, for example, FIG. 24 and column 8, lines 55-62) based on a sequence of events and state transitions (see, for example, column 9, lines 11-14). The test specification is considered to be a script that describes the occurrence of events according to an input test sequence (see, for example, column 9, lines 48-64). Note that the tests account for the timing of each event (see, for example, column 4, lines 4-8). Yamauchi further discloses outputting the test specification or the script (see, for example, column 14, lines 14-25). Moreover, the states, events and transitions in the test sequence and in the test specification are output for display on a display section (see, for example, FIG. 4). The operation specification table, or the state transition matrix, as described above, is likewise output for display (see, for example, FIG. 2) according to position information (such as the "central coordinates" in FIG. 3).

Yamauchi further discloses the features of (h) above in terms of converting a state transition model describing events and actions into a data structure, stored in memory, having event and state identification codes (see, for example, FIG. 3 and column 9, lines 11-39). Yamauchi further discloses specifying, based on the model, a sequence of transitions that serves as a test sequence (see, for example, column 9, lines 48-56), and outputting a test specification that is based, in part, on the test sequence (see, for example, FIG. 1). In other words, Yamauchi discloses rewriting the test sequence into a test specification, which is to say rewriting information for pseudo-generating events (such as the test sequence in FIG. 5) into information corresponding to the events (such as the test specification in FIG. 24).

With respect to claim 4 (original), Takuma does not expressly disclose a script editor for editing said script file based on any one of an event input to be occurred, an occurrence timing of said event and an occurrence frequency.

However, Yamauchi further discloses the features above in terms of an editor for editing the test specification or script based on event inputs to occur (see, for example, FIG. 8 and column 10, lines 43-48).

It would have been obvious to one of ordinary skill in the art at the time the invention was made to incorporate the features taught by Yamauchi into the system of Takuma, so as to enable the testing of programs based on a plurality of events and state transitions (see, for example, Yamauchi, column 3, lines 62-67). The modification would have been obvious because one of ordinary skill in the art would have been motivated to facilitate debugging in the Takuma system (see, for example, Takuma, column 3, lines 33-36).

With respect to claim 5 (original), Takuma does not expressly disclose the limitation wherein said script file is any one of a timing chart format, a text format and a message sequence chart format.

However, Yamauchi further discloses the features above in terms of a test specification or script that is in a text format (see, for example, FIG. 24 and column 14, lines 5-8).

It would have been obvious to one of ordinary skill in the art at the time the invention was made to incorporate the features taught by Yamauchi into the system of Takuma, so as to enable the testing of programs based on a plurality of events and state transitions (see, for example, Yamauchi, column 3, lines 62-67). The modification would have been obvious because one of ordinary skill in the art would have been motivated to facilitate debugging in the Takuma system (see, for example, Takuma, column 3, lines 33-36).

With respect to claims 6, 7 and 8 (original), Takuma also discloses the limitation wherein said program includes a main routine for executing a main process of said system and a normal generating event routine for normally generating a corresponding event based on various data and a signal transmitted from said other component to said first central processing unit (see, for example, column 7, lines 37-50, which shows a processor having a routine for executing the instructions of a main monitor program and for responding to, i.e. generating events corresponding to, signals sent from a host processor).

With respect to claims 9, 10 and 11 (currently amended), Takuma does not expressly disclose the limitation wherein said event pseudo-generating information is information of a generating technique in accordance with said event.

However, Yamauchi further discloses the features above in terms of a test sequence data structure, i.e. event pseudo-generating information, which comprises a sequential arrangement of events and actions (see, for example, column 9, lines 57-64).

It would have been obvious to one of ordinary skill in the art at the time the invention was made to incorporate the features taught by Yamauchi into the system of Takuma, so as to enable the testing of programs based on a plurality of events and state transitions (see, for example, Yamauchi, column 3, lines 62-67). The modification would have been obvious because one of ordinary skill in the art would have been motivated to facilitate debugging in the Takuma system (see, for example, Takuma, column 3, lines 33-36).

Takuma does not expressly disclose the limitation:

wherein said program includes: a) a main routine for executing main processes of the system, and b) an event normal generating routine for, based on a signal supplied from the system due to operator selection performed on the system, detecting the operator selection and notifying said main routine of the operator selection.

However, Yamauchi further discloses the features above in terms of an editor for detecting selections made by an operator and notifying the editor to update the display accordingly (see, for example, FIG. 2 and column 9, lines 11-25). The system inherently supplies signals based on the operator's interaction with the system, such as by moving the cursor.

Takuma in view of Yamauchi also discloses the limitation:

wherein said pseudo-generating routine is automatically generated in a programming language that is the same as or similar to a programming language of said main routine (see, for example, column 7, lines 37-50, which shows that the same VLIW processor executes both the instructions that compose the debug target program, i.e. the generated program, and the instructions that compose the main monitor program, i.e. the main routine, which is to say that the programming language is the same or is similar for both).

With respect to claims 12, 13 and 14 (original), Takuma does not expressly disclose the limitation wherein said event is any one of a message-type for receiving a start message from another task or another apparatus, a flag-type for reading a variation of a variable or an input/output, an interrupt-type for receiving an interrupt from an outside, an in-mail type for notifying an internal event which occurs in a cell of said state-transition matrix to another state-transition matrix when said state-transition matrix is layered and a function-call type for calling a function executing a group of processes.

However, Yamauchi further discloses that the events include internal events and external events (see, for example, column 3, lines 21-45). External events that cause state transitions in response to external operations (see, for example, column 10, lines 27-35) are considered to be interrupt-type events "for receiving an interrupt from an outside."

It would have been obvious to one of ordinary skill in the art at the time the invention was made to incorporate the features taught by Yamauchi into the system of Takuma, so as to enable the testing of programs based on a plurality of events and state transitions (see, for example, Yamauchi, column 3, lines 62-67). The modification would have been obvious because one of ordinary skill in the art would have been motivated to facilitate debugging in the Takuma system (see, for example, Takuma, column 3, lines 33-36).

With respect to claim 15 (currently amended), the claim recites a program development method that corresponds to the program development apparatus recited in claim 1 (see Takuma and Yamauchi as applied to claim 1 above).

With respect to claim 16 (currently amended), the claim recites a program development method that corresponds to the program development apparatus recited in claim 2 (see Takuma and Yamauchi as applied to claim 2 above).

With respect to claim 17 (currently amended), the claim recites a program development method that corresponds to the program development apparatus recited in claim 3 (see Takuma and Yamauchi as applied to claim 3 above).

With respect to claim 18 (original), the claim recites a program development method that corresponds to the program development apparatus recited in claim 4 (see Takuma and Yamauchi as applied to claim 4 above).

With respect to claim 19 (original), the claim recites a program development method that corresponds to the program development apparatus recited in claim 5 (see Takuma and Yamauchi as applied to claim 5 above).

With respect to claims 20, 21 and 22 (original), the claims recite program development methods that correspond to the program development apparatus recited in claims 6, 7 and 8, respectively (see Takuma and Yamauchi as applied to claims 6, 7 and 8 above).

With respect to claims 23, 24 and 25 (original), the claims recite program development methods that correspond to the program development apparatus recited in claims 9, 10 and 11, respectively (see Takuma and Yamauchi as applied to claims 9, 10 and 11 above).

With respect to claims 26, 27 and 28 (original), the claims recite program development methods that correspond to the program development apparatus recited in claims 12, 13 and 14, respectively (see Takuma and Yamauchi as applied to claims 12, 13 and 14 above).

With respect to claim 29 (currently amended), the claim recites a program development program that corresponds to the program development apparatus recited in claim 1 (see Takuma and Yamauchi as applied to claim 1 above).

With respect to claim 30 (currently amended), the claim recites a storage medium storing a program development program that corresponds to the program development apparatus recited in claim 1 (see Takuma and Yamauchi as applied to claim 1 above).

With respect to claim 31 (currently amended), the claim recites a program development program that corresponds to the program development apparatus recited in claim 2 (see Takuma and Yamauchi as applied to claim 2 above).

Conclusion

11. The prior art made of record and not relied upon is considered pertinent to Applicant's disclosure. U.S. Pat. No. 5,995,739 to Rotbart discloses a device for implementing hierarchical state charts and methods and apparatus useful therefor. U.S. Pat. No. 6,006,028 to Aharon et al.

discloses a test program generator. U.S. Pat. No. 6,654,715 to Iwashita discloses an apparatus, method and storage medium for verifying a logical device.

12. Applicant's amendment necessitated the new ground(s) of rejection presented in this Office action. Accordingly, **THIS ACTION IS MADE FINAL**. See MPEP § 706.07(a). Applicant is reminded of the extension of time policy as set forth in 37 CFR 1.136(a).

A shortened statutory period for reply to this final action is set to expire THREE MONTHS from the mailing date of this action. In the event a first reply is filed within TWO MONTHS of the mailing date of this final action and the advisory action is not mailed until after the end of the THREE-MONTH shortened statutory period, then the shortened statutory period will expire on the date the advisory action is mailed, and any extension fee pursuant to 37 CFR 1.136(a) will be calculated from the mailing date of the advisory action. In no event, however, will the statutory period for reply expire later than SIX MONTHS from the date of this final action.

13. Any inquiry concerning this communication or earlier communications from the examiner should be directed to Michael J. Yigdall whose telephone number is (571) 272-3707. The examiner can normally be reached on Monday through Friday from 7:30am to 4:00pm.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Tuan Q. Dam can be reached on (571) 272-3695. The fax phone number for the organization where this application or proceeding is assigned is 703-872-9306.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free).

MY

Michael J. Yigdall
Examiner
Art Unit 2122

mjy

TUAN DAM
SUPERVISORY PATENT EXAMINER